The Graph Signature:
A Scalable Query Optimization Index for RDF Graph Databases Using Bisimulation and Trace Equivalence Summarization

Mustafa Jarrar, Sina Institute, Birzeit University, Birzeit, Palestine
Anton Deik, Sina Institute, Birzeit University, Birzeit, Palestine

ABSTRACT

Querying large data graphs has brought the attention of the research community. Many solutions were proposed, such as Oracle Semantic Technologies, Virtuoso, RDF3X, and C-Store, among others. Although such approaches have shown good performance in queries with medium complexity, they perform poorly when the complexity of the queries increases. In this paper, the authors propose the Graph Signature Index, a novel and scalable approach to index and query large data graphs. The idea is that they summarize a graph and instead of executing the query on the original graph, they execute it on the summaries. The authors’ experiments with Yago (16M triples) have shown that e.g., a query with 4 levels costs 62 sec using Oracle but it only costs about 0.6 sec with their index. Their index can be implemented on top of any Graph database, but they chose to implement it as an extension to Oracle on top of the SEM.MATCH table function. The paper also introduces disk-based versions of the Trace Equivalence and Bisimilarity algorithms to summarize data graphs, and discusses their complexity and usability for RDF graphs.

Keywords: Big Data, Bisimulation, Data Index, Data Web, Graph Databases, Mashups, Query Optimization, RDF, RDF Stores, Structural Summaries, Trace Equivalence

1. INTRODUCTION AND MOTIVATION

Big Data, Data Web, Linked and Open Data are examples of an emerging era of data industry and data science. We are witnessing a rapid growth of the amount of available structured and linked data. As of June 2015, the Linking Open Data Statistics project (LODStats) records 3308 published datasets consisting of around 89.9 billion RDF triples. Examples of published datasets are: DBPedia, Yago, DBLP, CiteSeer, ACM, Freebase, Geonames, MusicBrainz, as well as open governmental datasets such as those of the UK (data.gov.uk), the US (data.gov), and Ireland (opendata.ie), and many others. The biggest software companies are encouraging the trend of publishing and linking structured data on the web. For example, Google, Yahoo, and Microsoft

DOI: 10.4018/IJSWIS.2015040102
have joined efforts in developing a shared ontology. Facebook is also providing access to parts of its data via its Graph API.

To exploit structured data on the web to its full potential, people need efficient querying methods. SPARQL was introduced by W3C as a standardized query language that enables querying decentralized collections of RDF data. However, SPARQL is oriented for technical people. So, in order to allow people with limited IT skills to query structured data, many solutions were proposed, among which are those which proposed an interactive approach that allows the user to formulate queries without prior knowledge of the underlying data or its structure. Examples of such approaches are: Lore (Goldman & Widom, 1997) which was developed for querying schema-free XML, and MashQL, which is a query formulation language for RDF introduced in previous work (Jarrar & Dikaiakos, 2008; 2009; 2010; 2012). In July 2013, Facebook started rolling out a “Graph Search” functionality, allowing users to formulate structured queries over the Facebook data graph. Not only do such approaches motivate the importance of querying data graphs, but they also emphasize the significance of having fast responses for queries executed over large data graphs in an interactive environment.

The most widely adopted data model specification for representing structured data on the web is RDF (Resource Description Framework). RDF syntax is based on XML and reflexes simple graph-based data model. RDF represents data as triples \(<\text{Subject}, \text{Predicate}, \text{Object}>\). For instance, the fact that the book called \textit{Wamadat} is authored by \textit{Naima} can be represented by the following three triples which form a directed labeled graph (see Figure 1):

\[
\begin{align*}
\langle BK3, Name, Wamadat \rangle \\
\langle BK3, Author, AU3 \rangle \\
\langle AU3, Name, Naima \rangle
\end{align*}
\]

Data representation using RDF is more elementary than relational databases and XML models, which enables easy data integration and interoperability of systems. However, querying RDF data graphs (especially large graphs) is a major challenge that faces all querying approaches, and therefore has brought the attention of the research community (e.g., Abadi et al., 2007; Chong et al., 2005; Schätzle et al., 2013; Tran et al., 2013; Yuan et al., 2013). Querying such data, which is typically stored in one relational table denoted by \(<S,P,O>\) is of high complexity because traversing a graph stored in relational model involves many self-joins of that table. More specifically, a query with \((n)\) edges on such a table requires \((n-1)\) self joins of that table (Abadi et al., 2007).

Several solutions were proposed to solve the problem of querying large RDF datasets. Among these solutions are Oracle Semantic Technologies (Chong et al., 2005), OpenLink Virtuoso (Erling & Mikhailov, 2007; 2010), Vertical Partitioning (Abadi et al., 2007), and RDF3X (Neumann & Weikum, 2008), to name a few. Although these approaches have shown good performance in queries with medium complexity (such queries cost several seconds), they tend to perform poorly when the complexity of the queries increase. In this paper, we provide an optimization solution for a type of queries that these solutions tend to perform poorly in. Specifically, we propose an index called the Graph Signature Index (GS). The idea of our index is to summarize the RDF graph through grouping nodes based on their outgoing and their incoming paths, and instead of executing the query on the original data graph, we execute it on the index. Because the resultant summaries are typically much smaller than the original graph, executing the query on the index is faster. Our proposed index can be implemented on top of any of the aforementioned solutions. However, we choose here to demonstrate our work on top of Oracle Semantic Technologies because of its availability and support by one of the most reputable Database Management Engines in the world (Oracle Database). Our experiments have shown a promising enhancement; e.g., a...
Related Content

Techniques for Named Entity Recognition: A Survey
www.igi-global.com/chapter/techniques-named-entity-recognition/65694?camid=4v1a

A Comparative Analysis of Online Social Networking Sites and Their Business Models
www.igi-global.com/chapter/comparative-analysis-online-social-networking/39197?camid=4v1a
Expressive OWL Queries: Design, Evaluation, Visualization
www.igi-global.com/article/expressive-owl-queries/75774?camid=4v1a

Sarcastic Sentiment Detection Based on Types of Sarcasm Occurring in Twitter Data
www.igi-global.com/article/sarcastic-sentiment-detection-based-on-types-of-sarcasm-occurring-in-twitter-data/189766?camid=4v1a